

WHAT IS CLAIMED IS:

1. A visual linker, comprising:
- an allocation module for allocating sections of
code and data into memory of a processor;
- an output module for writing a file of said
allocated sections of code and data;
- an incomplete link module, wherein said
incomplete link comprises allocation information
on those sections that are allocated by said
allocation module and those that have not yet
been allocated;
- a link server that interprets linking
instructions modifies said incomplete link
module accordingly and invokes said allocation
module accordingly and invokes said output
module accordingly;
- an interface for receiving instructions for said
link server and for providing feedback as to the
state of said incomplete link;
- and a graphical user interface that generates
said instructions in response to user gestures
and graphically displays the state of said
incomplete link.
2. A method of incrementally and interactively
allocating code and data sections, comprising
the steps of:
- generate a specific allocation instruction from
a client program or program component;
- executing said instruction by making alterations
to allocation information associated with one or

more code or data sections;
resolving allocation to the full extent possible
given the current allocation information
associated with all code and data sections
involved in the link;
report to client programs the current allocation
state inclusive of allocation errors and
sections not yet allocated;
and repeating these steps until all sections of
code and data have been allocated.

3. The method of Claim 2 including the step of
interpreting a user gesture made to a graphical
user interface as a specific allocation
instruction.
4. The method of Claim 3 wherein a gesture is a
drag-and-drop operation or a point-and-click
operation on a video screen.
5. The method of Claim 2 including the step of
displaying said current allocation state
graphically to the user, inclusive of allocation
errors and sections not yet allocated.
6. The method of Claim 2 including the step of
writing the results to an output file.
7. The method of Claim 2 including the step of
recording said instructions for replay in a
linking strategy file.
8. The method of Claim 7 wherein the record of
instructions may be displayed and altered
through a graphical user interface.
9. The method of Claim 2 wherein the set of code
and data sections involved in the link may be

determined by a reachability check performed using a cross-reference graph.

10. The method of Claim 2 wherein an instruction may specify an overflow policy to be used whenever a portion of memory fills up during allocation.

11. The method of Claim 2 wherein an instruction may provide for an allocator-optimized stack or heap size by specifying a minimum and maximum size instead of a particular size.

12. The method of Claim 2 wherein one instruction may apply allocation operations to a related group of sections simultaneously.

13. The method of Claim 12 wherein the sections may be the set of sections reachable from a specific starting section, as determined by a cross-reference graph.